



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09179851 A**(43) Date of publication of application: **11.07.97**

(51) Int. Cl.

**G06F 17/12**(21) Application number: **07336710**(22) Date of filing: **25.12.95**(71) Applicant: **FUJITSU LTD**(72) Inventor: **ORII SHIGEO  
KOMATSU HIDEMI**(54) **PARALLEL PROCESSING SYSTEM**

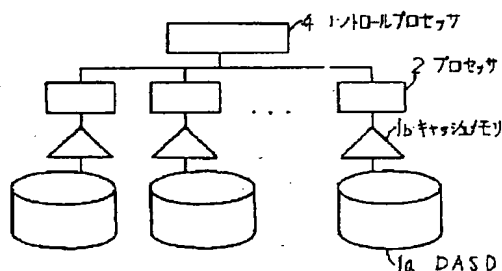
for the processing to another processor.

(57) Abstract:

COPYRIGHT: (C)1997,JPO

**PROBLEM TO BE SOLVED:** To solve a large scale of simultaneous linear equation which is generated at the time of analyzing the structure of an automobile, etc., at high speed and also with simple configuration through the use of plural processors and DASD, etc., by dividing a coefficient matrix into plural storage means and executing a processing in parallel.

**SOLUTION:** The storage means (DASD1a and a cache memory 1b) stores information indicating a pivot, information indicating a block, the matrix element of the prescribed matrix which belongs to the block and information indicating the element. A processing means (the processor 2) executes an erasing processing with the whole elements except the pivot in a column which the pivot belongs to as zero as against the respective elements in the row which the respective elements belong to at every prescribed column based on information obtained from the storage means and a transfer means through the use of the matrix element of the row which the pivot belongs to. The transfer means (the processor 2) transfers information indicating the elements together with pivot and the matrix element which is used





(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-179851

(43) 公開日 平成9年(1997)7月11日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 17/12

識別記号

庁内整理番号

F I

G 0 6 F 15/324

技術表示箇所

審査請求 未請求 請求項の数 4 O L (全 8 頁)

(21) 出願番号 特願平7-336710

(22) 出願日 平成7年(1995)12月25日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72) 発明者 折居 茂夫

神奈川県川崎市中原区上小田中1015番地

富士通 株式会社内

(72) 発明者 小松 秀実

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74) 代理人 弁理士 井桁 貞一

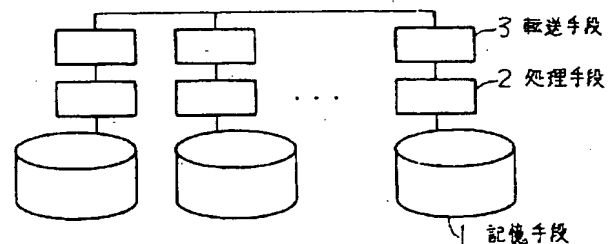
(54) 【発明の名称】 並列処理システム

(57) 【要約】

【課題】 本発明は、自動車等の構造解析を行う際に生ずる大規模な連立一次方程式を、複数のプロセッサ、D A S D等を用いて、高速かつ簡潔な構成で解くことを目的とする。

【解決手段】 当該課題を解決するために、係数行列をブロック単位に分割し、枢軸を示す情報、東海ブロックを示す情報、当該ブロックに属する所定の行数の行列要素と当該要素を示す情報を記憶する記憶手段と、当該記憶手段と後述の転送手段から得られる情報に基づいて、枢軸の属する行の行列要素を用いて、枢軸の属する列の枢軸以外の全要素を零とする消去処理を当該各要素の属する行の各要素に対し所定の行ごとに行う処理手段と、枢軸と当該処理に用いる行列要素と共にそれらの要素を示す情報を他のプロセッサに転送する転送手段とから構成する。

本発明の構成を示す原理図



## 【特許請求の範囲】

【請求項 1】連立一次方程式の係数行列と定数ベクトルからなる対象行列の行列要素をブロック単位に複数の記憶手段に分割して、連立一次方程式を解く並列処理システムにおいて、

枢軸を示す情報、当該ブロックを示す情報、当該ブロックに属する所定の行数の行列要素と当該要素を示す情報を記憶する記憶手段と、

当該記憶手段と後述の転送手段から得られる情報に基づいて、枢軸の属する行の行列要素を用いて、枢軸の属する列の枢軸以外の全要素を零とする消去処理を当該各要素の属する行の各要素に対し所定の列ごとに行う処理手段と、

枢軸と当該処理に用いる行列要素と共にそれらの要素を示す情報を他のプロセッサに転送する転送手段とを有することを特徴とする並列処理システム。

【請求項 2】連立一次方程式の係数行列と定数ベクトルからなる対象行列の行列要素をブロック単位に複数の記憶手段に分割して、連立一次方程式を解く並列処理システムにおいて、

枢軸を示す情報、当該ブロックを示す情報、当該ブロックに属する所定の列数の行列要素と当該要素を示す情報を記憶する記憶手段と、

当該記憶手段と後述の転送手段から得られる情報に基づいて、枢軸の属する行の行列要素を用いて、枢軸の属する列の枢軸以外の全要素を零とする消去処理を当該各要素の属する行の各要素に対し所定の行ごとに行う処理手段と、

枢軸と当該処理に用いる行列要素と共にそれらの要素を示す情報を他のプロセッサに転送する転送手段とを有することを特徴とする並列処理システム。

【請求項 3】請求項 1 記載の並列処理システムにおいて、請求項 1 記載の処理手段は、請求項 1 記載の記憶手段から、列ブロックごとに当該ブロックに属する行列要素

$$\sum_j A(i, j) x(j) = b(i) \quad (1)$$

である。まず、一番目の式の  $(-A(i, 1)/A(1, 1))$  倍を  $i$  番目の式に加える ( $i=2 \sim N$ )。すると、 $i$  番目の式の  $x(1)$  の係数を 0 にすることができ、 $i$  番目の式の  $A(i, j)$ 、 $b(i)$  は、 $A(i, j) = A(i, j) - A(i, 1) \cdot A(1, j)/A(1, 1)$ 、 $b(i) = b(i) - A(i, 1) b(1)/A(1, 1)$  と更新される ( $j=1 \sim N$ )。ここで、 $A(i, 1) = 0$ 、( $i=2 \sim N$ ) となり、方程式 (1) の二番目の式以降は、 $x(2)$ 、 $\dots$ 、 $x(N)$  に関する  $(N-1)$  元の連立一次方程式となり、問題のサイズを一つ小さくすることができる。次に、同様にして、二番目の式の  $(-A(i, 2)/A(2, 2))$  倍を  $i$  番目の式に加えると、 $A(i, 2) = 0$  ( $i=3 \sim N$ ) となり、 $x$

素と当該要素を示す情報を第二の記憶手段に読出し、列ブロックごとに処理することを特徴とする請求項 1 記載の並列処理システム。

【請求項 4】請求項 2 記載の並列処理システムにおいて、請求項 2 記載の処理手段は、請求項 2 記載の記憶手段から、行ブロックごとに当該ブロックに属する行列要素と当該要素を示す情報を第二の記憶手段に読出し、行ブロックごとに処理することを特徴とする請求項 2 記載の並列処理システム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、自動車、建設、電機等の分野において、構造解析を行う際に生ずる大規模な連立一次方程式を、並列処理システムを用いて、高速にかつ簡単な構成で解く並列処理システムに関する。

## 【0002】

【従来の技術】従来、有限要素法や差分法等により生ずる連立一次方程式の係数行列の規模が計算機のメモリに入りきらない程大きい場合、係数行列のデータの分割による制御の複雑さを回避するため、係数行列の全データを、単一の外部記憶装置に格納し、処理していた。この場合、処理時間短縮のため、複数の CPU により並列に処理し、当該処理に必要なデータを、当該外部記憶装置からメモリに読出し、適当なブロックごとに処理していた。

【0003】また、通常、このような連立一次方程式を解く場合、解法アルゴリズムとして、ガウスの消去法等が用いられる。これは、ガウスの消去法が、直接法であり、必ず解を得ることができるためである。ここでは、後の説明の便宜を考慮し、ガウスの消去法について説明する。解くべき  $N$  元の連立一次方程式を、 $Ax=b$  とする。ここで、 $A$  は係数行列、 $x$  は未知数ベクトル、 $b$  は定数ベクトルである。成分別書くと、

(2) を消去することができ、問題を、 $x(3)$ 、 $\dots$ 、 $x(N)$  に関する  $(N-2)$  元の連立一次方程式に縮減することができる。以降についても同様で、一般に  $x(k-1)$  が消去された段階においては、処理 P『 $i=k+1$  から  $N$  までに対し、 $b(i) = b(i) - A(i, k) \cdot b(k)/A(k, k)$ 、および  $j=1$  から  $N$  までに対し、 $A(i, j) = A(i, j) - A(i, k) \cdot A(k, j)/A(k, k)$ 』を行うことにより、 $x(k)$  が消去でき、問題は  $(N-k)$  元の連立一次方程式に帰着する。この  $k$  行目の消去に用いる非零の対角要素  $A(k, k)$  を枢軸という。処理 P を、 $k=1$  から  $k=N-1$  まで行くと、問題は、 $A(N, N) \cdot x(N) = b(N)$  という簡単な式に縮減される。ここまでの処理を前進消去という。前進消去が終了する

と、次は、後退代入という処理に入る。前進消去が終了した時点で(1)の方程式は、下三角部分の行列要素は0、上三角部分の行列要素のみが非零に変形されており、N番目の式は、簡単に解けて、 $x(N) = b(N) / A(N, N)$ となる。ただし、 $b(N)$ 、 $A(N, N)$ は、処理Pによる更新後の値である。得られた $x(N)$ を(N-1)番目の式に代入すると、 $x(N-1)$ が求まる。求まった $x(N)$ および $x(N-1)$ を(N-2)番目の式に代入すると、 $x(N-2)$ が求まる。このようにして、降順に、解 $x(i)$  ( $i=1 \sim N$ )を得ることができる。

【0004】

【発明が解決しようとする課題】上述のように、係数行列を分割せず、単一の外部記憶装置に格納したとすると、上述した並列処理によりCPUの処理時間は短縮できても、データの入出力時間は短縮されず、結果として、解を得るまでの時間が短縮されないという第一の問題があった。

【0005】また、ガウスの消去法は、解を得るために、前進消去と後退代入の2つの処理を行わなければならないため、対象行列の行列要素を複数の外部記憶装置に分割して処理する場合、プログラム、つまり制御が複雑になるという第二の問題があった。

【0006】

【課題を解決するための手段】本発明では、第二の問題を解決すべく、解法アルゴリズムとして、制御を簡単にするガウス—ヨルダン法を採用し、また、第一の問題を解決すべく、係数行列を適当なブロックで分割し、他のプロセッサとデータの通信を行うことにより、全体の処理時間の短縮を図る。具体的には、請求項1に係る発明の場合、枢軸を示す情報、当該ブロックを示す情報、当該ブロックに属する所定の行数の行列要素と当該要素を示す情報を記憶する記憶手段と、当該記憶手段と後述の転送手段から得られる情報に基づいて、枢軸の属する行の行列要素を用いて、枢軸の属する列の枢軸以外の全要素を零とする消去処理を当該各要素の属する行の各要素に対し所定の列ごとに行う処理手段と、枢軸と当該処理に用いる行列要素と共にそれらの要素を示す情報を他のプロセッサに転送する転送手段とから構成する。第1図は、本発明の構成の原理図を表す。同図中、1は方程式の行列要素と当該要素を特定する情報等を記憶した記憶手段を、2は記憶手段1と転送手段3から得られる情報に基づいて、枢軸の属する行の行列要素を用いて、枢軸の属する列の枢軸以外の全要素を零とする更新処理を当該各要素の属する行の各要素に対し所定の列ごとに行う処理手段を表す。

【0007】本発明で用いるガウス—ヨルダン法について、説明する。上述したガウスの消去法においては、処理Pから明らかなように、方程式を一回り小さく変形して解くべく、枢軸 $A(k, k)$ を用いて、第k列の第k

行より下の部分だけを消去の対象としていた。これを變形して全非対角要素を消去するアルゴリズムが、ガウス—ヨルダン法である。つまり、ガウス消去法の処理Pの全体に『 $i=1$ から $k-1$ 』の処理を加えると、ガウス—ヨルダン法のアルゴリズムが得られる。具体的に書き下すと、『 $i=1$ から $k-1$ および $k+1$ からNまでに対し、 $b(i) = b(i) - A(i, k) \cdot b(k) / A(k, k)$ 、および $j=1$ からNまでに対し、 $A(i, j) = A(i, j) - A(i, k) \cdot A(k, j) / A(k, k)$ 』となる。 $k=1$ からNまでについて、この処理が終了すると、全非対角要素は0となり、対角要素のみが残り、簡単に解 $x(i)$ を得ることができる( $i=1, \dots, N$ )。したがって、ガウス—ヨルダン法では、後退代入の必要がないため、プログラム、つまり制御が簡単になる。

【0008】

【実施の形態】本発明の実施の形態を、第2図から第8図に基づき説明する。なお、以下の説明は、請求項3に係る発明についてのものであるが、他の請求項についても基本的には同様である。第2図は、本発明の実施例を示す構成図である。同図中、1aはM個からなるDASDを、1bはM個からなるキャッシュメモリを、2はM個からなるプロセッサを、4はコントロール・プロセッサ(以下、「CP」という)を表す。ここで、DASD 1aおよびキャッシュメモリ1bは第1図の記憶手段1に、プロセッサ2は第1図の処理手段2および転送手段3に、それぞれ対応するものである。解くべき方程式としては、N元の連立一次方程式、 $Ax = b$ を考える。なお、 $A$ 、 $x$ 、 $b$ は、前出式(1)と同一対象を表す。まず、本発明の全体の処理フローを、第3図に基づき説明し、その後に各処理ステップの詳細を説明する。

【0009】CP 4は、係数行列 $A$ 、定数ベクトル $b$ からなる対象行列を、第4図に示すように、行ブロックごとに、各DASD 1aに分割する。CP 4は、各プロセッサ2に、担当するブロックを示す情報と枢軸の初期値 $A(1, 1)$ を示す情報とを送る。各プロセッサ2は、自配下のDASD 1aに格納された行ブロックを特定するための第5図に示すような情報を、主記憶に格納する。CP 4からキャッシュメモリへの読出開始信号を受けた各プロセッサ2は、自配下のDASD 1aから処理単位である第6図に示す列ブロックに属する行列要素を、キャッシュメモリ1bに読出す(S1)。キャッシュメモリ1bには、例えば、第6図に示すように、行列要素が格納される。また、上述したように、各プロセッサ2は、その主記憶に、第5図に示すように、担当ブロックと枢軸を示す情報を格納しており、当該情報に基づき、処理における枢軸が自配下のDASD 1aにあるか否かを判定する(S2)。

【0010】自配下に枢軸があるプロセッサは、他のプロセッサに対し、自配下のキャッシュメモリにある枢軸

を転送する。また、後で第9図を用いて説明するが、第j列を処理する場合、第kプロセッサ( $k=1 \sim j-1$ )は、他のプロセッサに対し、第j列の属するブロック内で、枢軸の属する第k行の行列要素を送信する(S3)。枢軸の属するプロセッサは、キャッシュメモリ内の行列要素を用いて、枢軸の属さないプロセッサは、当該転送されてきた行列要素と自配下のキャッシュメモリ内の行列要素を用いて、枢軸の属する列の枢軸以外の全行列要素を零とするガウス・ヨルダン法による消去処理を行う。各プロセッサ2は、各列の消去処理を終えると、当該処理の完了通知をCP4に送信し、CP4は、各プロセッサに次列の処理開始信号を送る。これを受けた各プロセッサ2は、主記憶内の枢軸指示情報を更新する。この処理を繰り返し、一の列ブロックに属する全枢軸について、前記消去処理を終了すると、各プロセッサ2は、CP4に対し、当該列ブロックの処理の完了を通知(S4)、DASDに更新内容を書込む。CP4は、全プロセッサから当該通知を受信すると、各プロセッサ2に対し、次の列ブロックの行列要素をキャッシュメモリに読み出すトリガとなる読出開始信号を送信する。

【0011】また、各プロセッサ2は、主記憶内の枢軸指示情報に基づき、処理対象の列ブロックが最終ブロックであるか否かを判定する(S5)。最終ブロックでないと判定された場合は、次の列ブロックに属する行列要素をキャッシュメモリに読み込む。最終ブロックであると判定された場合は、既に非対角要素は、零と変形されているため、解 $x(i)$ を、 $x(i) = b(i) / A(i, i)$ として求める(S6)。ここで、 $b(i)$ および $A(i)$ は、消去処理S4により更新された値である。以上が、本発明の処理の全体的なフローの概略である。

【0012】次に、各フローの詳細について、説明する。以下では、説明の簡単化のため、 $N=2M$ 、処理単位の列ブロックを2列として説明する。まず、対象行列の分割の態様を、第4図に基づき説明する。第4図において、第3図と同一符号を付したものは、第3図のものと同一の対象を示す。第4図において、5は、係数行列Aと定数ベクトルbからなる対象行列を表す。CP4は、対象行列の行列要素を、各DASDに均等となるよう割り当てる。この対象行列の各DASDへの分割については、今は、 $N=2M$ の場合を考えているから、第4図に示すように、行ブロックには、対象行列の各2行が割り当てられる。CP4は、各DASDに対し、分割された対象行列の行列要素を送信し、各プロセッサ2に対し、自配下にあるDASDに格納されている行ブロックを特定する情報を送信する。各プロセッサ2は、受信したこの情報を、第5図に示すように、その主記憶に格納する。DASDに格納している行ブロックを特定する情報としては、行単位で分割しているため、第5図に示す

ように、行のみを特定し、各プロセッサは、担当ブロックに、フラグをたて、担当でないブロックと区別している。また、各プロセッサは、処理を行うにあたって、現在どの段階まで消去処理をしたか、つまり枢軸の情報を知る必要があるため、主記憶は、第5図に示すように、枢軸を示す情報を格納している。各プロセッサは、CP4からの上述の処理開始信号または読出開始信号を受けて、主記憶内の枢軸指示情報を逐次更新する。第5図では、1行目に枢軸があること、つまり、枢軸はA(1, 1)であることを示している。

【0013】次に、処理単位である列ブロックを、第6図に基づき説明する。第6図は、対象行列において、処理単位である列ブロックを示した図である。ここでは、列ブロックの単位として2列を考えているため、第6図に示すように、一の列ブロックは2列の行列要素から構成されている。各プロセッサ2は、この列ブロック単位に、キャッシュメモリ1bにその行列要素を読出し、消去処理S4を行う。

【0014】次に、CP4からの上述の読出開始信号の受信により、キャッシュメモリに読出されるデータ構造について、第7図に基づいて説明する。第7図は、行ブロック1担当のキャッシュメモリに格納される列ブロック1の行列要素を示す。各プロセッサ2は、自配下のDASDに格納された行列要素を列ブロックごとに読出するので、列ブロック1処理時の行ブロック1担当のキャッシュメモリに、第7図に示すように、A(1, 1)、A(1, 2)、A(2, 1)およびA(2, 2)の値を格納する。他のキャッシュメモリについても、同様である。

【0015】次に、枢軸以外のプロセッサ間のデータ通信を、第9図に基づいて説明する。第9図は、係数行列Aと定数ベクトルbからなる対象行列を、行ブロックおよび列ブロックに分割した図である。ブロックを(行ブロック、列ブロック)として特定すると、第9図に示すように、ブロック(k, k)は、枢軸の含まれるブロックを示す( $k=1 \sim M$ )。ここでは、行ブロックに分割し、列ブロック毎に処理を進める場合を考えているため、列ブロックkの消去処理を行う場合、枢軸の属する第jプロセッサは、他のプロセッサに対し、ブロック(j, k)内の枢軸の属する行の行列要素を送信する。例えば、列ブロック2の消去処理を行う場合、第1プロセッサは、他のプロセッサに対し、ブロック(1, 2)内の枢軸A(1, 1)の属する行列要素、A(1, 3)、A(1, 4)を送信する。各プロセッサは、転送されてきた行列要素を記憶しておき、各列ブロックの消去処理前に更新処理を行う。

【0016】各記憶手段が持つべきデータ等についての説明は終了したので、列ブロック1の行列要素がDASDからキャッシュメモリに読出された後の消去処理S4について、第10図に基づき説明し、そのあと、具体的な処理を説明する。列ブロックL( $L=1 \sim M$ )の消去

処理S4は、第10図に示すように、4つのステップからなる。まず、第一に、列ブロックJ ( $J=1 \sim L-1$ ) の消去処理による列ブロックLの行列要素の更新を、 $J=1, 2, \dots, L-1$ の順に行う(S41)。したがって、列ブロック1の消去処理においては、このステップはない。この更新処理を終えると、各プロセッサは、自メモリ内にある枢軸指示情報から特定される枢軸を用いて、ガウス・ヨルダンのアルゴリズムによる消去処理を行う(S42)。消去処理を終えると、各プロセッサは、列ブロック(L+1)~Mで使用する行列要素の更新処理を行う(S43)。各プロセッサは、CP4に対し、列ブロックLの処理が終了した旨

第1プロセッサ:

$$A(2, 1) = A(2, 1) / A(1, 1)$$

$$A(2, 2) = A(2, 2) - A(2, 1) \cdot A(1, 2)。$$

第2プロセッサ:

$$A(3, 1) = A(3, 1) / A(1, 1)$$

$$A(4, 1) = A(4, 1) / A(1, 1)$$

$$A(3, 2) = A(3, 2) - A(3, 1) \cdot A(1, 2)$$

$$A(4, 2) = A(4, 2) - A(4, 1) \cdot A(1, 2)。$$

第3プロセッサ:

...

第Mプロセッサ:

$$A(N-1, 1) = A(N-1, 1) / A(1, 1)$$

$$A(N, 1) = A(N, 1) / A(1, 1)$$

$$A(N-1, 2) = A(N-1, 2) - A(N-1, 1) \cdot A(1, 2)$$

$$A(N, 2) = A(N, 2) - A(N, 1) \cdot A(1, 2)。$$

ここで、第iプロセッサとは、行ブロックi担当のプロセッサをいう( $i=1 \sim M$ )。各プロセッサは、列ブロック1の第1列の消去処理が終わったことを、CP4に対し通知する。CP4は、全プロセッサからその通知を受けると、各プロセッサに対し、第2列の消去処理を開始する信号を送信する。その信号を受けた各プロセッサ2は、枢軸指示情報をA(2, 2)を示すように更新し、枢軸A(2, 2)を、他のプロセッサに転送する。各プロセッサ2は、既に枢軸A(2, 2)の値は変形済であるため、次の列ブロックで使用する行列要素の更新処理、

第1プロセッサ:

$$A(1, 2) = A(1, 2) / A(2, 2)$$

第2プロセッサ:

$$A(3, 2) = A(3, 2) / A(2, 2)$$

$$A(4, 2) = A(4, 2) / A(2, 2)$$

第3プロセッサ:

...

第Mプロセッサ:

$$A(N-1, 2) = A(N-1, 2) / A(2, 2)$$

$$A(N, 2) = A(N, 2) / A(2, 2)$$

を行う。この処理が終了すると、各プロセッサ2は、列ブロック1の処理が終了したことを、CP4に対し通知

の通知を送信し(S44)、列ブロックLの消去処理が終わると共に、処理後の内容を、各DASDに書込む。

【0017】次に、第10図に示す消去処理にしたがって、消去処理を具体的に説明する。まず、各プロセッサ2は、主記憶内の枢軸情報を参照し、自キャッシュメモリに枢軸があるか否かを判定する。現在、第5図に示すように、枢軸情報は「1」を示している。したがって、第1行担当のプロセッサ1は、自キャッシュメモリ内の行列要素A(1, 2)およびA(1, 1)を、他のプロセッサに転送する。その後、各プロセッサは、以下の消去処理を行う。

し、CP4は、各プロセッサ2に対し、処理の終了した列ブロック1の行列要素をDASDに書き込み、次の列ブロックをキャッシュメモリに読み込むよう読出開始信号を送信する。その信号を受信した各プロセッサ2は、DASDへの書込処理、キャッシュメモリへの読込処理を行う。また、前記読出開始信号を受信した各プロセッサ2は、自メモリ内の枢軸指示情報を、A(3, 3)を示すように更新する。各プロセッサ2は、その枢軸指示情報を参照し、転送する行列要素を決定する。具体的には、自配下のキャッシュメモリに枢軸の属する第2プロセッサは、その属する枢軸A(3, 3)を転送し、枢軸指示情報よりその番号の小さい第1プロセッサは、列ブロック2に属する自配下のキャッシュメモリにあるA(1, 3)、A(1, 4)、A(2, 3)およびA(2, 4)を特定する情報とその値を、他のプロセッサに転送する(第9図参照)。受信した各プロセッサは、それ以降の列ブロック処理で生じる更新処理に用いるため、これらの転送された行列要素をそのキャッシュメモリに記憶しておく。

【0018】各プロセッサ2は、列ブロック2についての消去処理を行う前に、列ブロック1の消去処理による更新処理を行う。具体的には、各プロセッサは、対象行列の列ブロック1の第1列の消去処理による更新処理:

第1プロセッサ:

$$A(2, 3) = A(2, 3) - A(2, 1) \cdot A(1, 3)$$

$$A(2, 4) = A(2, 4) - A(2, 1) \cdot A(1, 4)$$

第2プロセッサ:

$$A(3, 3) = A(3, 3) - A(3, 1) \cdot A(1, 3)$$

$$A(4, 3) = A(4, 3) - A(4, 1) \cdot A(1, 3)$$

$$A(3, 4) = A(3, 4) - A(3, 1) \cdot A(1, 4)$$

$$A(4, 4) = A(4, 4) - A(4, 1) \cdot A(1, 4)$$

第3プロセッサ:

...

第Mプロセッサ:

$$A(N-1, 3) = A(N-1, 3) - A(N-1, 1) \cdot A(1, 3)$$

$$A(N, 3) = A(N, 3) - A(N, 1) \cdot A(1, 3)$$

$$A(N-1, 4) = A(N-1, 4) - A(N-1, 1) \cdot A(1, 4)$$

$$A(N, 4) = A(N, 4) - A(N, 1) \cdot A(1, 4)$$

を行う。また、各プロセッサは、対象行列の列ブロック 1 の第2列の消去処理に伴う更新処理:

第1プロセッサ:

$$A(1, 3) = A(1, 3) - A(1, 2) \cdot A(2, 3)$$

$$A(1, 4) = A(1, 4) - A(1, 2) \cdot A(2, 4)$$

第2プロセッサ:

$$A(3, 3) = A(3, 3) - A(3, 2) \cdot A(2, 3)$$

$$A(4, 3) = A(4, 3) - A(4, 2) \cdot A(2, 3)$$

$$A(3, 4) = A(3, 4) - A(3, 2) \cdot A(2, 4)$$

$$A(4, 4) = A(4, 4) - A(4, 2) \cdot A(2, 4)$$

第3プロセッサ:

...

第Mプロセッサ:

$$A(N-1, 3) = A(N-1, 3) - A(N-1, 2) \cdot A(2, 3)$$

$$A(N, 3) = A(N, 3) - A(N, 2) \cdot A(2, 3)$$

$$A(N-1, 4) = A(N-1, 4) - A(N-1, 2) \cdot A(2, 4)$$

$$A(N, 4) = A(N, 4) - A(N, 2) \cdot A(2, 4)$$

を行う。これで、列ブロック1の消去処理による列ブロック2の行列要素の更新処理を終わり、列ブロック2の非対角要素の消去処理を行う。各プロセッサは、以上のようにして、CP4の制御の下、キャッシュメモリに列ブロックごとに行列要素を読み込み、ガウス・ヨルダン法による消去処理を進めていく。

【0019】

【発明の効果】本発明によると、係数行列を複数の記憶手段に分割し、並列に処理するため、従来、計算時間の大部分を占めていた入出力時間を、大幅に短縮（理論的には、記憶手段の数をMとすると、処理時間は $1/M$ に短縮）することができる。これは、解法アルゴリズムとして、ガウス・ヨルダン法を用いることにより、プログラムを簡素化し、分割されたデータの扱いを簡易なものとすることができたことに起因する。本発明においては、行列要素を分割して持つことができるため、従来よ

り、大きな規模の連立一次方程式を解くことが可能となる。また、本発明では、行ブロックまたは列ブロックごとにまとめて処理を行うため、列または行ごとに行う処理より、高速に処理できるという効果もある。

【図面の簡単な説明】

【図1】 本発明の構成を示す原理図

【図2】 本発明の実施例を示す構成図

【図3】 本発明の処理フローを示す図

【図4】 対象行列の分割の態様を示す図

【図5】 枢軸と担当ブロックを示すデータ構造図

【図6】 処理単位である列ブロックを示す図

【図7】 キャッシュメモリ上の行列要素のデータ図

【図8】 行列要素 $A(i, j)$ の更新処理に必要な行列要素を示す図

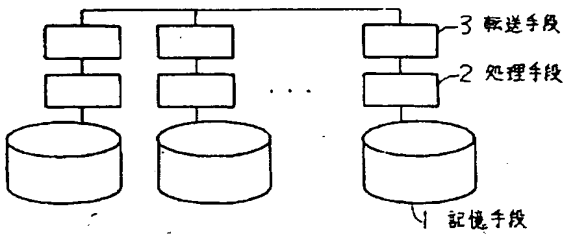
【図9】 転送する行列要素を示す図

【図10】 消去処理S4を示す図



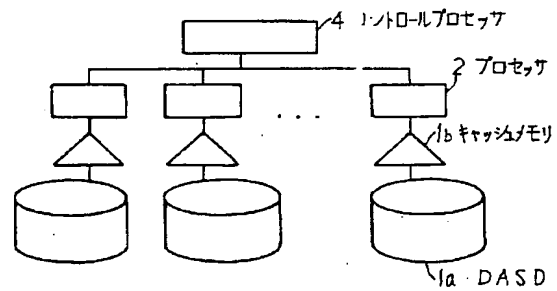
【図1】

本発明の構成を示す原理図



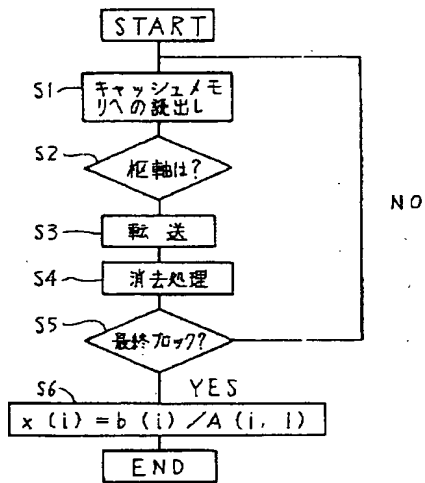
【図2】

本発明の実施例を示す構成図



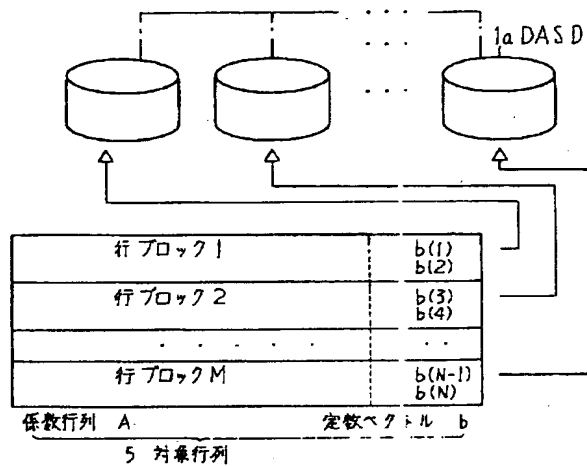
【図3】

本発明の処理フローを示す図



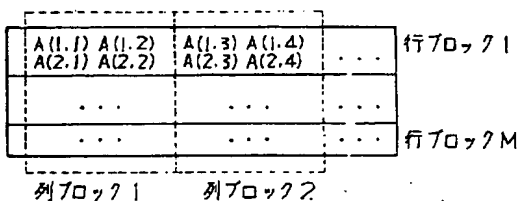
【図4】

対象行列の分割の様様を示す図



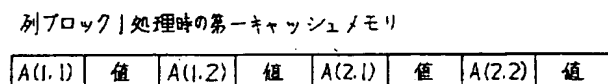
【図6】

処理単位である列ブロックを示す図



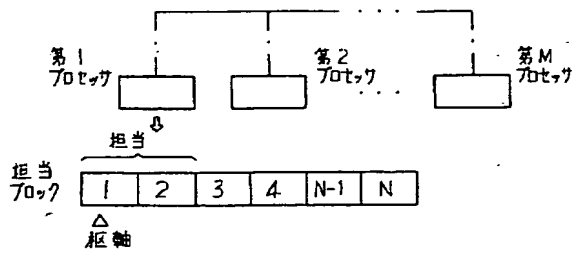
【図7】

キャッシュメモリ上の行列要素のデータ図



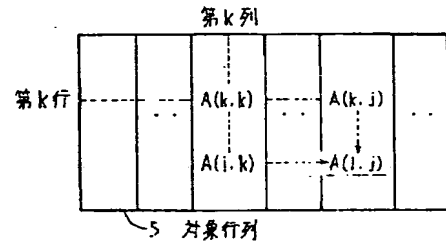
【図5】

枢軸と担当ブロックを示すデータ構造図



【図8】

$A(i, j)$  の更新処理に必要な行列要素を示す図



【図9】

転送する行列要素を示す図

枢軸が  $A(j, j)$  の場合

列ブロック $j/2-1$		列ブロック $j/2$ (= 処理対象)	
$A(j-4, j-2)$	$A(j-4, j-1)$	$A(j-4, j)$	$A(j-4, j+1)$
$A(j-3, j-2)$	$A(j-3, j-1)$	$A(j-3, j)$	$A(j-3, j+1)$
$A(j-2, j-2)$	$A(j-2, j-1)$	$A(j-2, j)$	$A(j-2, j+1)$
$A(j-1, j-2)$	$A(j-1, j-1)$	$A(j-1, j)$	$A(j-1, j+1)$
$A(j, j-2)$	$A(j, j-1)$	$A(j, j)$	$A(j, j+1)$
$A(j+1, j-2)$	$A(j+1, j-1)$	$A(j+1, j)$	$A(j+1, j+1)$
$A(j+2, j-2)$	$A(j+2, j-1)$	$A(j+2, j)$	$A(j+2, j+1)$
$A(j+3, j-2)$	$A(j+3, j-1)$	$A(j+3, j)$	$A(j+3, j+1)$

【図10】

消去処理 (S4) を示す図

